

Amendment and Response

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM**IN THE CLAIMS**

Please amend claim 1 as follows:

1. (Currently Amended) A method of managing memory in a distributed computer system, the method comprising:

binding a remote key to a first address representing a contiguous memory address range accessible by a first consumer process at a first endnode;

sending the bound remote key and first address from the first endnode to a second endnode on a communication fabric; and

performing a remote direct memory access operation with a second consumer process from the second endnode to access the contiguous memory address range including sending the bound remote key and the first address from the second endnode to the first endnode on the communication fabric.

2. (Previously Presented) The method of claim 1 further comprising:

controlling local memory access protection in the first endnode with a virtual memory manager in an operating system kernel process.

3. (Previously Presented) The method of claim 1 further comprising:

comparing the bound remote key and the corresponding first address supplied by the second endnode to the bound remote key and corresponding first address in the first endnode.

4. (Previously Presented) The method of claim 3 wherein if the bound remote key and corresponding first address supplied by the second endnode do not match the bound remote key and first address in the first endnode, the second endnode is not granted access to the contiguous memory address range.

5. (Previously Presented) The method of claim 3 wherein if the contiguous memory address range represented by the first address bound to the remote key supplied by the second endnode is invalid, the second endnode is not granted access to the contiguous memory address range.

Amendment and Response

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

6. (Previously Presented) The method of claim 1 wherein the first address is an effective address pointing to an address space in memory accessible by a consumer process.
7. (Previously Presented) The method of claim 6 wherein the effective address points to a virtual address space.
8. (Previously Presented) The method of claim 7 further comprising:
comparing the bound remote key and the corresponding first virtual address supplied by the second endnode to the bound remote key and corresponding first virtual address in the first endnode; and
handling a page fault condition in the first endnode caused by the first virtual address bound to the remote key supplied by the second endnode not being previously mapped by an operating system of the first endnode.
9. (Previously Presented) The method of claim 6 wherein the consumer process is a user process.
10. (Previously Presented) The method of claim 6 wherein the consumer process is a kernel process.
11. (Previously Presented) The method of claim 1 wherein the first address is a virtual address accessible by a consumer kernel process.
12. (Previously Presented) The method of claim 1 wherein the binding includes associating the first address to the remote key with a consumer process employing a bind remote key verb.
13. (Previously Presented) The method of claim 1 further comprising:
obtaining at least one remote key with a consumer process employing an allocate remote key verb.

Amendment and Response

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

14. (Previously Presented) The method of claim 1 further comprising:
unbinding the remote key from the first address with a consumer process employing an unbind remote key verb.
15. (Previously Presented) The method of claim 13 further comprising:
retiring at least one remote key that was previously obtained via the allocate remote key verb with the consumer process employing a deallocate remote key verb.
16. (Previously Presented) The method of claim 1 wherein the remote key cannot be used to protect more than one memory region at a given instant.
17. (Previously Presented) The method of claim 1 further comprising:
reusing the remote key after the remote direct memory access operation from the second endnode is completed.
18. (Previously Presented) The method of claim 1 further comprising:
disabling a translation for the remote key after the remote key is used for the remote direct memory access operation from the second endnode.
19. (Previously Presented) A distributed computer system comprising:
communication fabric;
a first endnode including:
a first consumer process which binds a remote key to a first address
representing a contiguous memory address range accessible by the consumer process;
and
a first interface controller sending the bound remote key and first address from the first endnode on the communication fabric; and
a second endnode including:
a second interface controller receiving the bound remote key and first address from the first endnode via the communication fabric; and

Amendment and Response

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

a second consumer process performing a remote direct memory access operation from the second endnode to access the contiguous memory address range including sending, via the second interface controller, the bound remote key and the first address from the second endnode to the first endnode on the communication fabric.

20. (Previously Presented) The distributed computer system of claim 19 wherein first endnode further includes:

an operating system kernel process having a virtual memory manager controlling local memory access protection in the first endnode.

21. (Previously Presented) The distributed computer system of claim 19 wherein in the remote direct memory operation if the bound remote key and corresponding first address supplied by the second endnode do not match the bound remote key and first address in the first endnode, the second endnode is not granted access to the contiguous memory address range.

22. (Previously Presented) The distributed computer system of claim 19 wherein in the remote direct memory operation if the contiguous memory address range represented by the first address bound to the remote key supplied by the second endnode is invalid, the second endnode is not granted access to the contiguous memory address range.

23. (Previously Presented) The distributed computer system of claim 19 wherein the first address is an effective address pointing to an address space in memory accessible by the consumer process.

24. (Previously Presented) The distributed computer system of claim 23 wherein the effective address points to a virtual address space.

25. (Previously Presented) The distributed computer system of claim 19 wherein the consumer process is a user process.

Amendment and Response

Applicant: Michael R. Krause et al.

Serial No.: 09/980,920

Filed: April 11, 2002

Docket No.: 10002166-2

Title: MEMORY MANAGEMENT IN DISTRIBUTED COMPUTER SYSTEM

26. (Previously Presented) The distributed computer system of claim 19 wherein the consumer process is a kernel process.
27. (Previously Presented) The distributed computer system of claim 26 wherein the first address is a virtual address accessible by the consumer kernel process.
28. (Previously Presented) The distributed computer system of claim 19 wherein the consumer process employs a bind remote key verb to bind the first address to the remote key.
29. (Previously Presented) The distributed computer system of claim 19 wherein the consumer process employs an allocate remote key verb to obtain at least one remote key.
30. (Previously Presented) The distributed computer system of claim 19 wherein the consumer process employs an unbind remote key verb to unbind the remote key from the first address.
31. (Previously Presented) The distributed computer system of claim 29 wherein the consumer process employs a deallocate remote key verb to retire at least one remote key that was previously obtained via the allocate remote key verb.
32. (Previously Presented) The distributed computer system of claim 19 wherein the remote key cannot be used to protect more than one memory region at a given instant.
33. (Previously Presented) The distributed computer system of claim 19 wherein the consumer process reuses the remote key after the remote direct memory access operation from the second endnode is completed.
34. (Previously Presented) The distributed computer system of claim 19 wherein the remote key is a one-shot validating mechanism such that a translation for the remote key is disabled after the remote key is used for the remote direct memory access operation from the second endnode.